

APPLICATION  
FOR  
UNITED STATES LETTERS PATENT

TITLE: HINTED STEM PLACEMENT ON HIGH-RESOLUTION  
PIXEL GRID

APPLICANT: TERENCE S. DOWLING AND R. DAVID ARNOLD

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EL 572 620 515 US

December 15, 2000  
Date of Deposit

## HINTED STEM PLACEMENT ON HIGH-RESOLUTION PIXEL GRID

### BACKGROUND OF THE INVENTION

The present invention relates to the rendering of characters for display on devices having grayscale pixels. Grayscale pixels are pixels capable of displaying a number of tones, typically from pure light to pure dark.

United States Patent No. 5,943,063 ("the '063 patent") describes the creation of anti-aliased characters for a computer output device such as a cathode ray tube (CRT) monitor or a liquid crystal display (LCD) device. In particular, the '063 patent, the disclosure of which is incorporated here by reference, describes techniques for placing hinted character stems on a high-resolution grid that is later down-sampled for anti-aliased text rendering on a coarse grid – that is, a grid coarser than the high-resolution grid – of grayscale pixels.

### SUMMARY OF THE INVENTION

The invention provides improved apparatus and methods of hinted stem placement on a device pixel grid.

In general, in one aspect, the invention provides a new hinted stem placement policy that is referred to as an unbiased-stems policy. This policy places a hinted stem so that it marks the smallest possible number of full device pixels with the least possible movement of the hinted stem center.

In general, in another aspect, the invention provides a new hinted stem placement policy that is referred to as a black-edge policy. This policy places a hinted stem such that at least one edge of a stem that covers more than one device pixel is on a device pixel boundary. This avoids drawing a stem with, for example, two gray pixels. Each stem is adjusted separately so that it has at least one black edge. This is accomplished with as small a move as possible, either left or right, for a vertical stem, or up or down, for a horizontal one.

The invention can be implemented to realize one or more of the following advantages. The new policies introduce less distortion than do prior art policies. The black-edge policy ensures that if more than one coarse pixel is marked across a stem, there

Next, the process aligns stems of the character based upon the policy dictated by the client and by the grid ratio specified by the client (step 80). The process of the invention differs from the prior art in this step, where it implements two new policies that advantageously produce characters at small character sizes having both good contrast and low distortion. The policy can be selected dynamically by the renderer or the client, or it can be predetermined by the creator of the renderer.

After the stems of the character have been aligned, controls are set for resolution-dependent properties based on the grid ratio (step 82). For example, device dependent property adjustments, such as spot size (ratio of apparent pixel size to physical pixel size), can be handled in this step. Finally, a high-resolution bitmap is provided to the client (step 84). Typically, the client manages the grayscale rendering (that is, the down sampling) of the high-resolution bitmap, although such grayscale rendering can also be accomplished within the rendering process itself as mentioned previously. Examples of high-resolution bitmaps created in accordance with the invention and otherwise are shown in FIGS. 4-6, which will be described later.

FIG. 2 illustrates step 78 of FIG. 1 in greater detail. A grayscale font renderer 88 receives a call 89 from a client 90. The call includes a character pointer, a character size, a grid ratio, and a policy. The renderer 88 creates a call 91 to a parsed font program 92, which returns a character program (to create the font outline of the character), font values, and font family values 93. The renderer 88 then passes a high-resolution bitmap 95 to the client 90, which does the grayscale rendering and, typically, sends a low resolution pixel map 96 to an output device 94. The parsed font program can be a program such as described in the book Adobe Type 1 Font Format, Version 1.1 ("Type 1 Format"), available from Adobe Systems Incorporated of San Jose, California.

FIG. 3 illustrates step 80 of FIG. 1 in greater detail. First, the font matrix and stem hints (i.e., stem hint values) are received from the parsed font program 92 (step 100). This data is described in the Type 1 Format reference. The font matrix applies to the entire font and so may be received only once. When a scaling, rotation, or other transformation is to be applied, a transformation matrix is applied to the font matrix to yield a current matrix; otherwise, the font matrix is the current matrix. The current matrix is used to transform font units to fine cell units. Next, the renderer transforms the stem hints with the current matrix to

obtain (i) device grid locations of the stem edges and (ii) stem widths (step 102). The locations and widths are calculated with high precision. In the present context, a stem is either horizontal or vertical. As a consequence, stem alignment of the present invention does not affect the appearance of diagonal stems, except incidentally as result of a diagonal stem's being attached to vertical or horizontal stems that have been affected.

Then, each stem is processed as will now be described.

If the stem is a “ghost” stem (“yes” branch from decision step 114) , it is aligned to the coarse grid according to its ghost stem information (step 125). Ghost stems are described in Type 1 Format and, briefly, are stems that have only one edge to control.

Otherwise, the processing proceeds according to the policy (decision step 104).

If the policy is “hard-edge” or “soft-edge”, the process proceeds as described in the ‘063 patent. Under the hard-edge policy, vertical and horizontal stems are made perfectly black, i.e., grayscale is not used, and only curves and diagonals are subject to grayscale-type anti-aliasing. The soft-edge policy, on the other hand, permits anti-aliasing of the horizontal and vertical stems. Hard-edge policy tends to make sharper, darker stems, but may suffer from quantization effects. Soft-edge policy generates a smoother stem, but it may seem fuzzy to some viewers.

If the hard-edge policy is chosen, the process rounds the stem width to an integral coarse grid size (step 106), that is, to an integer multiple of a coarse grid cell width. Next, the stem is aligned to the coarse grid (step 108), which means the width of the stem completely fills one or more pixels, and the processing of the stem is complete (step 110).

If any other policy is selected (“other” branch from decision step 104), the process rounds the stem width to an integral multiple of the high-resolution grid cell size (step 112). Then, if the other policy is no policy, the processing of the stem is complete.

If the soft-edge policy is selected (“soft-edge” branch from decision step 200), the stem is aligned to the coarse grid so that the outside stem edge and a coarse grid line are aligned (step 202). The stem is moved the minimum distance necessary, if any, in either direction perpendicular to its edges, to achieve this alignment. A stem hint defines two edges, and the outside stem edge is the one that is farthest away from the center of the character. To simplify processing, an approximate value can be used for the center rather

than calculating an exact center. If the center of the stem passes through the center of the character, either edge may be chosen for alignment.

It should be noted that a reference to a stem edge in this specification in general refers to an edge as defined by a stem hint. A stem hint defines two parallel edges. Thus, a stem edge, as defined by a stem hint, will be straight even when the character outline of the stem is rounded, as in the case of the left and right sides of the letter “O”.

If the unbiased-stems policy is chosen (“unbiased stems” branch from decision step 200), the stem spread is considered – that is, how many coarse grid cells (each of which corresponds to a device pixel) are marked or painted by the stem (decision step 210). If the stem width is expressed in units of coarse grid cell width, so that a stem having a width of 1.6 is 1.6 times as wide as a coarse grid cell, then the minimum number of coarse grid cells a stem can paint is found by rounding the stem width up to the next whole number. Thus, a stem of width 1.6 can paint no fewer than two coarse grid cells. If the stem paints a minimum number of coarse grid cells, the processing of the stem is complete (step 110). Otherwise, the stem is moved the minimum distance necessary, in either direction perpendicular to its edges, so that the stem only paints the minimum necessary number of coarse grid cells (step 212). The movement is made to position the stem so that its edges align with the fine (high-resolution) grid. (Recall that the stem width is the width of an integral number of fine grid cells.) The processing of the stem is complete (step 110).

If the black-edge policy is chosen (“black-edge” branch from decision step 200), the stem width is considered (decision step 220). If the stem is not at least one full coarse grid cell (that is, one full device pixel) wide, the process proceeds to decision step 210.

Otherwise, the process moves the stem, if necessary, the minimum distance required, in either direction, so that at least one stem edge aligns with a coarse grid line. The processing of the stem is complete (step 110).

The black-edge and unbiased-stems policies are new. It may be noted that the prior art soft-edge policy and the new black-edge policy are, in a sense, extensions of the unbiased-stems policy. All three policies mark the minimum number of coarse pixels. The unbiased-stems policy does not move the stem except to achieve minimum coarse pixel coverage. The black-edge policy does the same thing, but may move the stem a bit farther to

get one stem edge aligned with the coarse grid. And the soft-edge policy goes even further in that it moves the stem to get a specific stem edge aligned with the coarse grid.

The client can select policies that are each specific to a particular stem. More typically, the client selects a policy for horizontal stems and possibly a different policy for vertical stems.

FIG. 4 shows the character “u” as delivered to a client (transfer of bitmap 95, FIG. 2) when no policy is in effect for vertical stems. (To simplify the illustration, the characters were hinted in y-direction, to align them to the baseline and x-height.) Vertical stems are accurately placed (5.25 pixels apart), but this results in gray-black-gray or gray-gray pixel combinations that appear softer than gray-black. The grid ratio is 4 so the character appears at four different positions or phases of 0, 1, 2, and 3 fine cells (40, 41, 42, and 43) relative to the coarse pixel grid. The coarse pixels or cells are shown in outline; the fine cells that are marked are shown as gray squares. The unmarked fine cells are shown implicitly.

FIG. 5 shows the same character as delivered when the black-edge policy is in effect for vertical stems. The grid ratio is 4 and the four phases 50, 51, 52, and 53 are shown. As is shown, the stems are aligned to the nearest device pixel edge, shifting either left or right. This reduces the shape distortion while maintaining sharp stems. Vertical stems are 5 or 5.5 pixels apart. In general, stems with a width of  $n.5$  pixels (that is, having a fractional part that is 0.5 device pixels) move in steps of zero or 0.5 pixels as phase is increased. Stems with a width of  $n.25$  or  $n.75$  pixels move in steps of zero, 0.25 or 0.75 pixels; and stems with an integral width move in steps of zero or one pixel, the same as with the soft-edge policy.

FIG. 6 shows the same character as delivered when the prior-art soft-edge policy is in effect for vertical stems. As is shown, the stem widths are not adjusted, but they are aligned so outside edges are on device pixel boundaries. This is accomplished by making the rasterizer aware of the device pixel boundaries. This policy produces sharper stems (no double gray) but distorts the character. It can result in uneven spacing with adjacent characters. Vertical stems are 5.5 or 6.5 pixels apart. In general, stems move in steps of zero or one pixel as phase is increased. All four phases 60, 61, 62, and 63 are shown, although phases 61, 62, and 63 are identical.

The invention can be implemented in digital electronic circuitry, or in computer hardware, firmware, software, or in combinations of them. Apparatus of the invention can be

implemented in a computer program product tangibly embodied in a machine-readable storage device for execution by a programmable processor; and method steps of the invention can be performed by a programmable processor executing a program of instructions to perform functions of the invention by operating on input data and generating output. The invention can be implemented advantageously in one or more computer programs that are executable on a programmable system including at least one programmable processor coupled to receive data and instructions from, and to transmit data and instructions to, a data storage system, at least one input device, and at least one output device. Each computer program can be implemented in a high-level procedural or object-oriented programming language, or in assembly or machine language if desired; and in any case, the language can be a compiled or interpreted language. Suitable processors include, by way of example, both general and special purpose microprocessors. Generally, a processor will receive instructions and data from a read-only memory and/or a random access memory. The essential elements of a computer are a processor for executing instructions and a memory. Generally, a computer will include one or more mass storage devices for storing data files; such devices include magnetic disks, such as internal hard disks and removable disks; magneto-optical disks; and optical disks. Storage devices suitable for tangibly embodying computer program instructions and data include all forms of non-volatile memory, including by way of example semiconductor memory devices such as EPROM, EEPROM, and flash memory devices; magnetic disks such as internal hard disks and removable disks; magneto-optical disks; and CD-ROM disks. Any of the foregoing can be supplemented by, or incorporated in, ASICs (application-specific integrated circuits).

The invention can be implemented in a computer system having a display device such as a monitor or LCD screen for displaying information to the user and a keyboard and a pointing device such as a mouse or a trackball by which the user can provide input to the computer system. The computer system can be programmed to provide a graphical user interface through which computer programs interact with users.

The invention has been described in terms of particular embodiments. Other embodiments are within the scope of the following claims. For example, steps of the invention can be performed in a different order and still achieve desirable results. Also, the description has implicitly assumed a down-sampling process that performs a simple mapping,

5           What is claimed is: